

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1998年 4月 3日

出 願 番 号

Application Number:

平成10年特許願第091844号

出 願 人

Applicant (s):

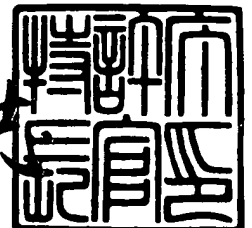
ソニー株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

1999年 2月26日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



【書類名】 特許願

【整理番号】 9800022505

【提出日】 平成10年 4月 3日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 15/00

【発明の名称】 グラフィック演算装置およびその方法

【請求項の数】 21

【発明者】

 【住所又は居所】 東京都品川区北品川 6丁目 7番 35号 ソニー株式会社
内

 【氏名】 黒瀬 悦和

【特許出願人】

 【識別番号】 000002185

 【氏名又は名称】 ソニー株式会社

 【代表者】 出井 伸之

【代理人】

 【識別番号】 100094053

 【弁理士】

 【氏名又は名称】 佐藤 隆久

【手数料の表示】

 【予納台帳番号】 014890

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9707389

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 グラフィック演算装置およびその方法

【特許請求の範囲】

【請求項 1】

ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置において、

同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断する画素位置判断手段と、

前記画素位置判断手段の判断結果に基づいて、前記同時に処理を行なおうとする複数の画素のうち、前記処理対象となっている単位図形の内側に位置する画素についての演算のみを実行する演算手段と

を有するグラフィック演算装置。

【請求項 2】

前記演算手段は、前記同時に処理を行なおうとする画素に対応する数の演算サブブロックをそれぞれ備えた直列に接続された複数の演算ブロックを有し、

前記複数の演算ブロックでパイプライン処理を実行する

請求項 1 に記載のグラフィック演算装置。

【請求項 3】

前記画素位置判断手段は、

前記演算手段にて処理されるデータに、前記判断の結果を示す有効性指示データを付加し、

前記演算手段は、前記有効性指示データに基づいて、当該有効性指示データに対応したデータを用いた演算処理を行なうか否かを決定する

請求項 1 に記載のグラフィック演算装置。

【請求項 4】

前記演算手段は、ディスプレイに表示する画素の R（赤），G（緑），B（青）データについて演算を行なう

請求項 1 に記載のグラフィック演算装置。

【請求項 5】

前記演算手段は、同次座標 (s, t) および同次項 q を用いた、テクスチャデータの読み出しのための演算を行なう

請求項 1 に記載のグラフィック演算装置。

【請求項 6】

ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置において、

前記単位図形に張り合わせるイメージデータであるテクスチャデータを記憶する記憶手段と、

前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成する画素位置判断手段と、

前記単位図形の頂点について、3次元座標 (x, y, z)、R (赤), G (緑), B (青) データ、同次座標 (s, t) および同次項 q を含むポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成手段と、

前記同時に処理を行なおうとする複数の画素のうち、前記有効性指示データが有効を示す画素の前記補間データに含まれる同次座標 (s, t) および同次項 q を用いて、前記記憶手段からテクスチャデータを読み出すテクスチャ処理手段とを有する

グラフィック演算装置。

【請求項 7】

前記記憶手段は、さらに z データと、ディスプレイに出力する表示データとを記憶し、

前記 z データを用いて、前記記憶手段から読み出したテクスチャデータが、記憶手段に既に記憶されている表示データよりも、視点側にあるか否かを判断し、

視点側にある場合には、前記記憶手段に記憶されている z データおよび表示データを更新する記憶制御手段

をさらに有する

請求項 6 に記載のグラフィック演算装置。

【請求項 8】

前記記憶制御手段は、前記読み出したテクスチャデータと、前記記憶手段に既に記憶されている表示データとを所定の混合比率で混合した表示データを生成し、当該混合した前記表示データで、前記記憶手段に既に記憶されている表示データを更新する

請求項 7 に記載のグラフィック演算装置。

【請求項 9】

前記記憶制御手段は、

複数の画素のデータについて、前記記憶手段へのアクセスを同時に行なうように制御する

請求項 6 に記載のグラフィック演算装置。

【請求項 10】

前記記憶手段は、DRAM である

請求項 6 に記載のグラフィック演算装置。

【請求項 11】

前記単位図形は、三角形である

請求項 6 に記載のグラフィック演算装置。

【請求項 12】

ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置において、

前記単位図形の頂点について、3次元座標 (x, y, z)、R (赤)、G (緑)、B (青) データ、同次座標 (s, t) および同次項 q を含むポリゴンレンダリングデータを生成するポリゴンレンダリングデータ生成装置と、

前記ポリゴンレンダリングデータを用いてレンダリング処理を行なうレンダリング装置と、

前記ポリゴンレンダリングデータ生成装置とレンダリング装置とを接続するバスと

を有し、

前記レンダリング装置は、

前記単位図形に張り合わせるイメージデータであるテクスチャデータを記憶する記憶手段と、

前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成する画素位置判断手段と、

前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成手段と、

前記同時に処理を行なおうとする複数の画素のうち、前記有効性指示データが有効を示す画素の前記補間データに含まれる同次座標 (s, t) および同次項 q を用いて、前記記憶手段からテクスチャデータを読み出すテクスチャ処理手段とを有する

グラフィック演算装置。

【請求項 13】

前記記憶手段は、さらに z データと、ディスプレイに出力する表示データとを記憶し、

前記 z データを用いて、前記記憶手段から読み出したテクスチャデータが、記憶手段に既に記憶されている表示データよりも、視点側にあるか否かを判断し、視点側にある場合には、前記記憶手段に記憶されている z データおよび表示データを更新する記憶制御手段

をさらに有する

請求項 12 に記載のグラフィック演算装置。

【請求項 14】

前記記憶制御手段は、前記読み出したテクスチャデータと、前記記憶手段に既

に記憶されている表示データとを所定の混合比率で混合した表示データを生成し、当該混合した前記表示データで、前記記憶手段に既に記憶されている表示データを更新する

請求項 13 に記載のグラフィック演算装置。

【請求項 15】

前記記憶制御手段は、

複数の画素のデータについて、前記記憶手段へのアクセスを同時に行なうように制御する

請求項 12 に記載のグラフィック演算装置。

【請求項 16】

前記記憶手段は、DRAMである

請求項 12 に記載のグラフィック演算装置。

【請求項 17】

前記単位図形は、三角形である

請求項 12 に記載のグラフィック演算装置。

【請求項 18】

ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算方法において、

同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、

前記判断の結果に基づいて、前記同時に処理を行なおうとする複数の画素のうち、前記処理対象となっている単位図形の内側に位置する画素についての演算のみを実行する

グラフィック演算方法。

【請求項 19】

前記演算は、ディスプレイに表示する画素の R（赤）、G（緑）、B（青）データについての演算である

請求項 18 に記載のグラフィック演算方法。

【請求項 20】

前記演算は、同次座標 (s, t) および同次項 q を用いた、テクスチャデータの読み出しのための演算である

請求項 18 に記載のグラフィック演算方法。

【請求項 21】

ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算方法において、

前記単位図形の頂点について、3次元座標 (x, y, z)、R (赤)、G (緑)、B (青) データ、同次座標 (s, t) および同次項 q を含むポリゴンレンダリングデータを生成し、

前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成し、

前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成し、

前記同時に処理を行なおうとする複数の画素のうち、前記有効性指示データが有効を示す画素の前記補間データに含まれる同次座標 (s, t) および同次項 q を用いて、前記記憶手段から、前記単位図形に張り合わせるイメージデータであるテクスチャデータを読み出す

グラフィック演算方法。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、高速演算を実現できるグラフィック演算装置およびその方法に関する。

【0002】

【従来の技術】

種々のCAD (Computer Aided Design) システムや、アミューズメント装置などにおいて、コンピュータグラフィックスがしばしば用いられている。特に、近年の画像処理技術の進展に伴い、3次元コンピュータグラフィックスを用いたシステムが急速に普及している。

このような3次元コンピュータグラフィックスでは、各画素（ピクセル）に対応する色を決定するときに、各画素の色の値を計算し、この計算した色の値を、当該画素に対応するディスプレイバッファ（フレームバッファ）のアドレスに書き込むレンダリング (Rendering) 処理を行う。

レンダリング処理の手法の一つに、ポリゴン (Polygon) レンダリングがある。この手法では、立体モデルを三角形の単位図形（ポリゴン）の組み合わせで表現し、このポリゴンを単位として処理を行い、描画することで、表示画面の色を決定する。

【0003】

ポリゴンレンダリングでは、物理座標系における三角形の各頂点についての、座標 (x, y, z) と、色データ (R, G, B, α) と、張り合わせのイメージパターンを示すテクスチャデータの同次座標 (s, t) および同次項 q の値とを入力とし、これらの値を三角形の内部で補間する処理が行われる。

ここで、同次項 q は、簡単にいうと、拡大縮小率のようなもので、実際のテクスチャバッファのUV座標系における座標、すなわち、テクスチャ座標データ (u, v) は、同次座標 (s, t) を同次項 q で除算した「 s/q 」および「 t/q 」に、それぞれテクスチャサイズ $U\ S\ I\ Z\ E$ および $V\ S\ I\ Z\ E$ を乗じたものとなる。

このような3次元コンピュータグラフィックシステムでは、例えば、ディスプレイバッファ（フレームバッファ）に描画を行う際に、画素毎に、テクスチャ座標データ (u, v) を用いてテクスチャバッファからテクスチャデータを読み出し、この読み出したテクスチャデータを、立体モデルの表面に三角形を単位として張り付けるテクスチャマッピング処理を行う。

なお、立体モデルでのテクスチャマッピングでは、各画素毎に、張り付けを行なうテクスチャデータが示す画像の拡大縮小率が変化する。

【0004】

ところで、このような3次元コンピュータグラフィックシステムでは、例えば、所定の矩形内の8画素についての処理を並行して（同時に）行う場合がある。

また、前述したような三角形を単位図形としたポリゴンレンダリングでは、張り付けを行うテクスチャデータの縮小率などは、三角形を単位として決定される。

従って、並行して処理を行った8画素分の演算結果のうち、対象となる三角形の外部に位置する画素についての演算結果は無効になる。

具体的には、図6に示すように、三角形30について所定の演算を行って縮小率を決定し、当該縮小率に応じたテクスチャデータを用いてテクスチャマッピング処理を行っている場合を考える。

ここで、矩形31、32、33は、それぞれ並行して処理される8（2×4）画素が配置された領域であり、ポリゴンレンダリング処理において、各矩形内に属する8画素については同じテクスチャデータが用いられる。

図6に示す場合には、矩形32に属する8画素は全て三角形30内に位置するため、8画素の演算結果は全て有効「1」である。これに対して、矩形31、33にそれぞれ属する8画素は、3画素は三角形30内に位置するが、5画素は三角形30の外に位置する。従って、8画素の演算結果のうち、3画素の演算結果は有効であるが、5画素の演算結果は無効となる。

従来では、矩形内に位置する8画素の全てについて、ポリゴンレンダリング処理を無条件に行っていた。

【0005】

【発明が解決しようとする課題】

しかしながら、上述したように、三角形を単位図形としたポリゴンレンダリング処理を行なう場合に、矩形内に位置する複数の画素の全てについての処理を、対象となっている三角形の内部に位置するか否かとは無関係に実行すると、膨大な数の無効な演算を行うことになり、消費電力に大きな影響を及ぼす。

特に、近年、3次元コンピュータグラフィックシステムの動作クロック周波数は非常に高くなっているため、消費電力の低下が大きな課題になっている。

【0006】

本発明は上述した従来技術の問題点に鑑みてなされ、消費電力の大幅な低下を図れるグラフィック演算装置およびその方法を提供することを目的とする。

【0007】

【課題を解決するための手段】

上述した従来技術の問題点を解決し、上述した目的を達成するために、本発明の第1の観点のグラフィック演算装置は、ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置であって、同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断する画素位置判断手段と、前記画素位置判断手段の判断結果に基づいて、前記同時に処理を行なおうとする複数の画素のうち、前記処理対象となっている単位図形の内側に位置する画素についての演算のみを実行する。

【0008】

本発明の第1の観点のグラフィック演算装置では、まず、画像位置判断手段において、同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かが判断される。

そして、演算手段では、前記画像位置判断手段の判断結果に基づいて、前記同時に処理を行なおうとする複数の画素のうち、前記処理対象となっている単位図形の内側に位置する画素についての演算が実行される。

一方、演算手段では、当該単位図形の外側に位置する画素についての演算は実行されない。

【0009】

また、本発明の第2の観点のグラフィック演算装置は、ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素に

についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置であって、前記単位図形に張り合わせるイメージデータであるテクスチャデータを記憶する記憶手段と、前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成する画素位置判断手段と、前記単位図形の頂点について、3次元座標(x, y, z)、R(赤), G(緑), B(青)データ、同次座標(s, t)および同次項qを含むポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成手段と、前記同時に処理を行なおうとする複数の画素のうち、前記有効性指示データが有効を示す画素の前記補間データに含まれる同次座標(s, t)および同次項qを用いて、前記記憶手段からテクスチャデータを読み出すテクスチャ処理手段とを有する。

【0010】

また、本発明の第3の観点のグラフィック演算装置は、ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置であって、前記単位図形の頂点について、3次元座標(x, y, z)、R(赤), G(緑), B(青)データ、同次座標(s, t)および同次項qを含むポリゴンレンダリングデータを生成するポリゴンレンダリングデータ生成装置と、前記ポリゴンレンダリングデータを用いてレンダリング処理を行なうレンダリング装置と、前記ポリゴンレンダリングデータ生成装置とレンダリング装置とを接続するバスとを有する。

ここで、前記レンダリング装置は、前記単位図形に張り合わせるイメージデータであるテクスチャデータを記憶する記憶手段と、前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成する画素位置判断手段と、前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成手段と、

前記同時に処理を行なおうとする複数の画素のうち、前記有効性指示データが有効を示す画素の前記補間データに含まれる同次座標 (s, t) および同次項 q を用いて、前記記憶手段からテクスチャデータを読み出すテクスチャ処理手段とを有する。

【0011】

また、本発明の第1の観点のグラフィック演算方法は、ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算方法であって、同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記判断の結果に基づいて、前記同時に処理を行なおうとする複数の画素のうち、前記処理対象となっている単位図形の内側に位置する画素についての演算のみを実行し、当該単位図形の外側に位置する画素についての演算を実行しない。

ここで、前記演算は、好適には、ディスプレイに表示する画素の R (赤), G (緑), B (青) データについての演算、あるいは、同次座標 (s, t) および同次項 q を用いた、テクスチャデータの読み出しのための演算である。

【0012】

さらに、本発明の第2の観点のグラフィック演算方法は、ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算方法であって、前記単位図形の頂点について、3次元座標 (x, y, z)、R (赤), G (緑), B (青) データ、同次座標 (s, t) および同次項 q を含むポリゴンレンダリングデータを生成し、前記同時に処理を行なおうとする複数の画素のそれぞれについて、前記単位図形の内側に位置するか否かを判断し、前記内側に位置する場合に有効を示す有効性指示データを生成し、前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成し、前記同時に処理を行なおうとする複数の画素のうち、前記有効

性指示データが有効を示す画素の前記補間データに含まれる同次座標 (s , t) および同次項 q を用いて、前記記憶手段から、前記単位図形に張り合わせるイメージデータであるテクスチャデータを読み出す。

【0013】

【発明の実施の形態】

以下、本実施形態においては、家庭用ゲーム機などに適用される、任意の3次元物体モデルに対する所望の3次元画像をCRT (Cathode Ray Tube) などのディスプレイ上に高速に表示する3次元コンピュータグラフィックシステムについて説明する。

図1は、本実施形態の3次元コンピュータグラフィックシステム1のシステム構成図である。

3次元コンピュータグラフィックシステム1は、立体モデルを単位図形である三角形(ポリゴン)の張り合わせとして表現し、このポリゴンを描画することで表示画面の各画素の色を決定し、ディスプレイに表示するポリゴンレンダリング処理を行うシステムである。

また、3次元コンピュータグラフィックシステム1では、平面上の位置を表現する(x , y)座標の他に、奥行きを表す z 座標を用いて3次元物体を表し、この(x , y , z)の3つの座標で3次元空間の任意の一点を特定する。

【0014】

図1に示すように、3次元コンピュータグラフィックシステム1は、メインメモリ2、I/Oインタフェース回路3、メインプロセッサ4およびレンダリング回路5が、メインバス6を介して接続されている。

以下、各構成要素の機能について説明する。

メインプロセッサ4は、例えば、ゲームの進行状況などに応じて、メインメモリ2から必要なグラフィックデータを読み出し、このグラフィックデータに対してクリッピング(Clipping)処理、ライティング(Lighting)処理およびジオメトリ(Geometry)処理などを行い、ポリゴンレンダリングデータを生成する。メインプロセッサ4は、ポリゴンレンダリングデータS4を、メインバス6を介してレンダリング回路5に出力する。

I/Oインタフェース回路3は、必要に応じて、外部からポリゴンレンダリングデータを入力し、これをメインバス6を介してレンダリング回路5に出力する。

【0015】

ここで、ポリゴンレンダリングデータは、ポリゴンの各3頂点の(x, y, z, R, G, B, α, s, t, q)のデータを含んでいる。

ここで、(x, y, z)データは、ポリゴンの頂点の3次元座標を示し、(R, G, B)データは、それぞれ当該3次元座標における赤、緑、青の輝度値を示している。

データαは、これから描画する画素と、ディスプレイバッファ21に既に記憶されている画素とのR, G, Bデータのブレンド(混合)係数を示している。

(s, t, q)データのうち、(s, t)は、対応するテクスチャの同次座標を示しており、qは同次項を示している。ここで、「s/q」および「t/q」に、それぞれテクスチャサイズUSIZEおよびVSIZEを乗じてテクスチャ座標データ(u, v)が得られる。テクスチャバッファ20に記憶されたテクスチャデータへのアクセスは、テクスチャ座標データ(u, v)を用いて行われる。

すなわち、ポリゴンレンダリングデータは、三角形の各頂点の物理座標値と、それぞれの頂点の色とテクスチャデータの同次座標および同次項を示している。

【0016】

以下、レンダリング回路5について詳細に説明する。

図1に示すように、レンダリング回路5は、DDA(Digital Differential Analyzer) セットアップ回路10、トライアングルDDA回路11、テクスチャエンジン回路12、メモリI/F回路13、CRTコントローラ回路14、RAMDAC回路15、DRAM16およびSRAM17を有する。

DRAM16は、テクスチャバッファ20、ディスプレイバッファ21、zバッファ22およびテクスチャCLUTバッファ23として機能する。

【0017】

DDAセットアップ回路10

DDAセットアップ回路10は、後段のトライアングルDDA回路11において物理座標系上の三角形の各頂点の値を線形補間して三角形の内部の各画素の色と深さ情報を求めるのに先立ち、ポリゴンレンダリングデータS4が示す(z , R , G , B , α , s , t , q)データについて、三角形の辺と水平方向の差分を求めるセットアップ演算を行う。

このセットアップ演算は、具体的には、開始点の値と終点の値と、開始点と終点との距離を用いて、単位長さ移動した場合における、求めようとしている値の変分を算出する。

【0018】

また、DDAセットアップ回路10は、並行して処理を行う8画素のそれぞれについて、処理対象となる三角形の内部に位置するか否かを示す1ビットの有効指示データ val を決定する。具体的には、有効指示データ val は、三角形の内部に位置する画素について「1」とし、三角形の外部に位置する画素について「0」とする。

DDAセットアップ回路10は、算出した変分データS10と、各画素の有効指示データ val とをトライアングルDDA回路11に出力する。

【0019】

トライアングルDDA回路11

トライアングルDDA回路11は、DDAセットアップ回路10から入力した変分データS10を用いて、三角形内部の各画素における線形補間された(z , R , G , B , α , s , t , q)データを算出する。

トライアングルDDA回路11は、各画素の(x , y)データと、当該(x , y)座標の画素についての(z , R , G , B , α , s , t , q , val)データとを、DDAデータ(補間データ)S11としてテクスチャエンジン回路12に出力する。

本実施形態では、トライアングルDDA回路11は、並行して処理を行う矩形内に位置する8画素分のDDAデータS11を単位としてテクスチャエンジン回路12に出力する。

【0020】

ここで、DDAデータS11の(z, R, G, B, α , s, t, q, val)データは、図2に示すように、161ビットのデータである。

具体的には、R, G, B, α データがそれぞれ8ビットであり、z, s, t, qデータがそれぞれ32ビットであり、valデータが1ビットである。

なお、以下、並行して処理を行う8画素についての(z, R, G, B, α , s, t, q, val)データのうち、valデータをvalデータS220₁～S220₈とし、(z, R, G, B, α , s, t, q)データを被演算データS221₁～S221₈とする。

すなわち、トライアングルDDA回路11は、8画素分の(x, y)データと、valデータS220₁～S220₈と、被演算データS221₁～S221₈とを単位としたDDAデータS11をテクスチャエンジン回路12に出力する。

【0021】

テクスチャエンジン回路12およびメモリI/F回路13

テクスチャエンジン回路12による、DDAデータS11を用いた、「s/q」および「t/q」の算出処理、テクスチャ座標データ(u, v)の算出処理、および、テクスチャバッファ20からの(R, G, B, α)データの読み出し処理と、メモリI/F回路13によるz比較処理および混合処理とを、図3に示す演算ブロック200, 201, 202, 204, 205でパイプライン方式で順に実行する。

ここで、演算ブロック200, 201, 202, 204, 205は、それぞれ8個の演算サブブロックを内蔵しており、8画素分の演算処理を並行して行う。

ここで、テクスチャエンジン回路12が演算ブロック200, 201, 202を内蔵し、メモリI/F回路13が演算ブロック204, 205を内蔵している。

【0022】

〔演算ブロック200〕

演算ブロック200は、DDAデータS11が示す(s, t, q)データを用

いて、sデータをqデータで除算する演算と、tデータをqデータで除算する演算とを行う。

演算ブロック200は、図3に示すように、8個の演算サブブロック $200_1 \sim 200_8$ を内蔵する。

ここで、演算サブブロック 200_1 は被演算データ $S221_1$ およびvalデータ $S220_1$ を入力し、valデータ $S220_1$ が「1」、すなわち有効であることを示す場合には、「 s/q 」および「 t/q 」を算出し、これを除算結果 $S200_1$ として演算ブロック201の演算サブブロック 201_1 に出力する。

【0023】

また、演算サブブロック 200_1 は、valデータ $S220_1$ が「0」、すなわち無効であることを示す場合には、演算は行わず、除算結果 $S200_1$ を出力しないか、あるいは、所定の仮値を示す除算結果 $S200_1$ を演算ブロック201の演算サブブロック 201_1 に出力する。

なお、演算サブブロック $200_2 \sim 200_8$ も、それぞれ対応する画素について、演算サブブロック 200_1 と同じ演算を行い、それぞれ除算結果 $S200_2 \sim S200_8$ およびvalデータ $S220_2 \sim S220_8$ を後段の演算ブロック201の演算サブブロック $201_2 \sim 201_8$ にそれぞれ出力する。

【0024】

図4は、演算サブブロック 200_1 の内部構成図である。

なお、図3に示す、全ての演算サブブロックは、基本的に、図4に示す構成をしている。

図4に示すように、演算サブブロック 200_1 は、クロックイネーブラ 210_1 、データ用フリップフロップ222、プロセッサエレメント223およびフラグ用フリップフロップ224を有する。

クロックイネーブラ 210_1 は、システムクロック信号 $S225$ を基準としたタイミングでvalデータ $S220_1$ を入力し、valデータ $S220_1$ のレベルを検出する。そして、クロックイネーブラ 210_1 は、valデータ $S220_1$ が、「1」である場合には、例えば、クロック信号 $S210_1$ にパルス発生させ、「0」である場合には、クロック信号 $S210_1$ にパルス発生させない。

【0025】

データ用フリップフロップ222は、クロック信号 $S210_1$ のパルスを検出すると、被演算データ $S221_1$ を取り込み、プロセッサエレメント223に出力する。

プロセッサエレメント223は、入力した被演算データ $S221_1$ を用いて前述した除算を行い、除算結果 $S200_1$ を演算サブブロック201₁のデータ用フリップフロップ222に出力する。

フラグ用フリップフロップ224は、システムクロック信号 $S225$ を基準としたタイミングで、 val データ $S220_1$ を取り込み、後段の演算ブロック201の演算サブブロック201₁のフラグ用フリップフロップ224に出力する。

なお、システムクロック信号 $S225$ は、図3に示す全ての演算サブブロック200₁～200₈，201₁～201₈，202₁～202₈，204₁～204₈のクロックイネーブラおよびフラグ用フリップフロップ224に供給される。

すなわち、演算サブブロック200₁～200₈，201₁～201₈，202₁～202₈，204₁～204₈における処理は同期して行われ、同一の演算ブロックに内蔵された8個の演算サブブロックは並行して処理を行う。

【0026】

〔演算ブロック201〕

演算ブロック201は、演算サブブロック201₁～201₈を有し、演算ブロック200から入力した除算結果 $S200_1$ ～ $S200_8$ が示す「 s/q 」および「 t/q 」に、それぞれテクスチャサイズ $USIZE$ および $VSIZE$ を乗じて、テクスチャ座標データ(u ， v)を生成する。

演算サブブロック201₁～201₈は、それぞれクロックイネーブラ211₁～211₈が val データ $S220_1$ ～ $S220_8$ のレベルを検出し、当該レベルが「1」の場合にのみ演算を行い、それぞれ演算結果であるテクスチャ座標データ $S201_1$ ～ $S201_8$ を、演算ブロック202の演算サブブロック202₁～202₈に出力する。

【0027】

〔演算ブロック202〕

演算ブロック202は、演算サブブロック $202_1 \sim 202_8$ を有し、メモリI/F回路13を介して、SRAM17あるいはDRAM16に、演算ブロック201で生成したテクスチャ座標データ(u, v)を含む読み出し要求を出力し、メモリI/F回路13を介して、SRAM17あるいはテクスチャバッファ20に記憶されているテクスチャデータを読み出すことで、(s, t)データに対応したテクスチャアドレスに記憶された(R, G, B, α)データS17を得る。

なお、テクスチャバッファ20には、MIPMAP（複数解像度テクスチャ）などの複数の縮小率に対応したテクスチャデータが記憶されている。ここで、何れの縮小率のテクスチャデータを用いるかは、所定のアルゴリズムを用いて、前記三角形を単位として決定される。

また、SRAM17には、テクスチャバッファ20に記憶されているテクスチャデータのコピーが記憶されている。

演算サブブロック $202_1 \sim 202_8$ は、それぞれクロックイネーブラ $212_1 \sim 212_8$ がvalデータ $S220_1 \sim S220_8$ のレベルを検出し、当該レベルが「1」の場合にのみ読み出し処理を行い、それぞれ読み出した(R, G,)データS17を、(R, G, B, α)データ $S202_1 \sim S202_8$ として、それぞれ演算ブロック203の演算サブブロック $203_1 \sim 203_8$ に出力する。

【0028】

なお、テクスチャエンジン回路12は、フルカラー方式の場合には、テクスチャバッファ20から読み出した(R, G, B, α)データを直接用いる。一方、テクスチャエンジン回路12は、インデックスカラー方式の場合には、予め作成したカラールックアップテーブル(CLUT)をテクスチャCLUTバッファ23から読み出して、内蔵するSRAMに転送および記憶し、このカラールックアップテーブルを用いて、テクスチャバッファ20から読み出したカラーインデックスに対応する(R, G, B)データを得る。

【0029】

〔演算ブロック203〕

演算ブロック203は、演算サブブロック $203_1 \sim 203_8$ を有し、演算ブロック202から入力したテクスチャデータである (R, G, B, α) データ $S202_1 \sim S202_8$ と、トライアングルDDA回路11からのDDAデータ $S11$ に含まれる (R, G, B) データとを、 (R, G, B, α) データ $S202_1 \sim S202_8$ に含まれる α データ（テクスチャ α ）が示す割合で混合し、 (R, G, B) 混合データを生成する。

そして、演算ブロック203は、生成された (R, G, B) 混合データと、対応するDDAデータ $S11$ に含まれる α データとを含む (R, G, B, α) データ $S203_1 \sim S203_8$ を、演算ブロック204に出力する。

【0030】

〔演算ブロック204〕

演算ブロック204は、演算サブブロック $204_1 \sim 204_8$ を有し、入力した (R, G, B, α) データ $S203_1 \sim S203_8$ について、zバッファ22に記憶されたzデータの内容を用いて、z比較を行い、 (R, G, B, α) データ $S203_1 \sim S203_8$ によって描画する画像が、前回、ディスプレイバッファ21に描画した値よりも手前（視点側）に位置する場合には、zバッファ22を更新すると共に、 (R, G, B, α) データ $S203_1 \sim S203_8$ を、 (R, G, B, α) データ $S204_1 \sim S204_8$ として、それぞれ演算ブロック205の演算サブブロック $205_1 \sim 205_8$ に出力する。

演算サブブロック $204_1 \sim 204_8$ は、それぞれクロックイネーブラ $214_1 \sim 214_8$ がvalデータ $S220_1 \sim S220_8$ のレベルを検出し、当該レベルが「1」の場合にのみz比較および (R, G, B, α) データ $S204_1 \sim S204_8$ の出力を行なう。

【0031】

〔演算ブロック205〕

演算ブロック205は、演算サブブロック $205_1 \sim 205_8$ を有し、入力した (R, G, B, α) データ $S204_1 \sim S204_8$ と、既にディスプレイバッ

ファ21に記憶されている(R, G, B)データとを、それぞれ(R, G, B, α)データS204₁～S204₈に含まれる α データが示す混合値で混合し、混合後の(R, G, B)データS205₁～S205₈をディスプレイバッファ21に書き込む(打ち込む)。

なお、メモリI/F回路13によるDRAM16に対してのアクセスは、16画素について同時に行なわれる。

演算サブブロック205₁～205₈は、それぞれクロックイネーブラ215₁～215₈がvalデータS220₁～S220₈のレベルを検出し、当該レベルが「1」の場合にのみディスプレイバッファ21に対しての書き込み処理を行う。

【0032】

CRTコントローラ回路14

CRTコントローラ回路14は、与えられた水平および垂直同期信号に同期して、図示しないCRTに表示するアドレスを発生し、ディスプレイバッファ21から表示データを読み出す要求をメモリI/F回路13に出力する。この要求に応じて、メモリI/F回路13は、ディスプレイバッファ21から一定の固まりで表示データを読み出す。CRTコントローラ回路14は、ディスプレイバッファ21から読み出した表示データを記憶するFIFO(First In First Out)回路を内蔵し、一定の時間間隔で、RAMDAC回路15に、RGBのインデックス値を出力する。

【0033】

RAMDAC回路15

RAMDAC回路15は、各インデックス値に対応するR, G, Bデータを記憶しており、CRTコントローラ回路14から入力したRGBのインデックス値に対応するデジタル形式のR, G, Bデータを、D/Aコンバータに転送し、アナログ形式のR, G, Bデータを生成する。RAMDAC回路15は、この生成されたR, G, BデータをCRTに出力する。

【0034】

以下、3次元コンピュータグラフィックシステム1の全体動作について説明す

る。

ポリゴンレンダリングデータ S4 が、メインバス 6 を介してメインプロセッサ 4 から DDA セットアップ回路 10 に出力され、DDA セットアップ回路 10 において、三角形の辺と水平方向の差分などを示す変分データ S10 が生成される。

この変分データ S10 は、トライアングル DDA 回路 11 に出力され、トライアングル DDA 回路 11 において、三角形内部の各画素における線形補間された $(z, R, G, B, \alpha, s, t, q)$ データが算出される。そして、この算出された $(z, R, G, B, \alpha, s, t, q)$ データと、三角形の各頂点の (x, y) データとが、DDA データ S11 として、トライアングル DDA 回路 11 からテクスチャエンジン回路 12 に出力される。

【0035】

次に、テクスチャエンジン回路 12 およびメモリ I/F 回路 13 において、DDA データ S11 を用いて、「 s/q 」および「 t/q 」の算出処理、テクスチャ座標データ (u, v) の算出処理、テクスチャバッファ 20 からのデジタルデータとしての (R, G, B, α) データの読み出し処理、混合処理、および、ディスプレイバッファ 21 への書き込み処理が、図 3 に示す演算ブロック 200, 201, 202, , 203, 204, 205 でパイプライン方式で順に実行される。

【0036】

次に、図 3 に示すテクスチャエンジン回路 12 およびメモリ I/F 回路 13 のパイプライン処理の動作について説明する。

ここでは、例えば、図 6 に示すような矩形 31 内の 8 画素について同時処理する場合を考える。この場合には、val データ $S220_1, S220_2, S220_3, S220_5, S220_6$ が「0」を示し、val データ $S220_4, S220_7, S220_8$ が「1」を示している。

【0037】

val データ $S220_1 \sim S220_8$ および被演算データ $S221_1 \sim S221_8$ が、それぞれ対応する演算サブブロック $200_1 \sim 200_8$ のクロックイネ

ープラ210₁～210₈に入力される。

そして、クロックイネープラ210₁～210₈において、それぞれvalデータS220₁～S220₈のレベルが検出される。具体的には、クロックイネープラ210₄，210₇，210₈において「1」が検出され、クロックイネープラ210₁，210₂，210₃，210₅，210₆において「0」が検出される。

その結果、演算サブブロック200₄，200₇，200₈においてのみ、被演算データS221₄，S221₇，S221₈を用いて、「s/q」および「t/q」が算出され、当該除算結果S200₄，S200₇，S200₈が演算ブロック201の演算ブロック201₄，201₇，201₈に出力される。

一方、演算サブブロック200₁，200₂，200₃，200₅，200₆では、除算は行なわれない。

また、除算結果S200₄，S200₇，S200₈の出力と同期して、valデータS220₁～S220₈が、演算ブロック201の演算サブブロック201₁～201₈に出力される。

【0038】

次に、演算サブブロック201₁～201₈のクロックイネープラ210₁～210₈において、それぞれvalデータS220₁～S220₈のレベルが検出される。

そして、この検出結果に基づいて、演算サブブロック201₄，201₇，201₈においてのみ、除算結果S200₄，S200₇，S200₈が示す「s/q」および「t/q」に、それぞれテクスチャサイズUSIZEおよびVSIZEを乗じて、テクスチャ座標データS202₄，S202₇，S202₈が生成され、それぞれ演算ブロック202の演算サブブロック202₄，202₇，202₈に出力される。

一方、演算サブブロック201₁，201₂，201₃，201₅，201₆では、演算は行なわれない。

また、テクスチャ座標データS202₄，S202₇，S202₈の出力と同期して、valデータS220₁～S220₈が、演算ブロック202の演算サ

ブブロック202₁ ~ 202₈ に出力される。

【0039】

次に、演算サブブロック202₁ ~ 202₈ のクロックイネーブラ212₁ ~ 212₈ において、それぞれvalデータS220₁ ~ S220₈ のレベルが検出される。

そして、この検出結果に基づいて、演算サブブロック202₄ , 202₇ , 202₈ においてのみ、SRAM17あるいはテクスチャバッファ20に記憶されているテクスチャデータの読み出し処理が行なわれ、(s, t)データに対応したテクスチャアドレスに記憶された(R, G, B, α)データが読み出される。

そして、この読み出した(R, G, B, α)データS202₄ , S202₇ , S202₈ が、演算ブロック204の演算サブブロック203₄ , 203₇ , 203₈ に出力される。

一方、演算サブブロック202₁ , 202₂ , 202₃ , 202₅ , 202₆ では、読み出し処理は行なわれない。

また、(R, G, B, α)データS202₄ , S202₇ , S202₈ の出力と同期して、valデータS220₁ ~ S220₈ が、演算ブロック203の演算サブブロック203₁ ~ 203₈ に出力される。

【0040】

次に、演算サブブロック203₁ ~ 203₈ のクロックイネーブラ212₁ ~ 212₈ において、それぞれvalデータS220₁ ~ S220₈ のレベルが検出される。

そして、この検出結果に基づいて、演算サブブロック203₄ , 203₇ , 203₈ においてのみ、それぞれ演算ブロック202から入力したテクスチャデータである(R, G, B, α)データS202₄ , 202₇ , 202₈ と、トライアングルDDA回路11からのDDAデータS11に含まれる(R, G, B)データとを、(R, G, B, α)データS202₄ , 202₇ , 202₈ に含まれるαデータ(テクスチャα)が示す割合で混合し、(R, G, B)混合データを生成する。

そして、演算サブブロック203₄ , 203₇ , 203₈ は、生成された(R

、G、B)混合データと、対応するDDAデータS11に含まれる α データとを含む(R、G、B、 α)データS203₄、203₇、203₈を、演算ブロック204に出力する。

一方、演算サブブロック203₁、203₂、203₃、203₅、203₆では、混合処理は行なわれない。

【0041】

次に、演算サブブロック204₁～204₈のクロックイネーブラ214₁～214₈において、それぞれvalデータS220₁～S220₈のレベルが検出される。

そして、この検出結果に基づいて、演算サブブロック204₄、204₇、204₈においてのみ、(R、G、B、 α)データS203₄、S203₇、S203₈について、zバッファ22に記憶されたzデータの内容を用いて、z比較が行なわれ、(R、G、B、 α)データS203₄、S203₇、S203₈によって描画する画像が、前回、ディスプレイバッファ21に描画した値よりも手前に位置する場合には、zバッファ22が更新されると共に、(R、G、B、 α)データS203₄、S203₇、S203₈が、それぞれ(R、G、B、 α)データS204₄、S204₇、S204₈として、それぞれ演算サブブロック205の演算サブブロック205₄、205₇、205₈に出力される。

【0042】

次に、演算サブブロック205₁～205₈のクロックイネーブラ215₁～215₈において、それぞれvalデータS220₁～S220₈のレベルが検出される。

そして、この検出結果に基づいて、(R、G、B、 α)データS204₄、S204₇、S204₈の(R、G、B)データと、既にディスプレイバッファ21に記憶されている(R、G、B)データとが、 α データが示す混合値で混合され、(R、G、B)データS205₄、S205₇、S205₈が最終的に算出される。

そして、この混合処理された、(R、G、B)データS205₄、S205₇、S205₈が、ディスプレイバッファ21に書き込まれる。

一方、演算サブブロック 204_1 、 204_2 、 204_3 、 204_5 、 204_6 では、混合処理は行なわれない。

【0043】

すなわち、テクスチャエンジン回路12およびメモリI/F回路13では、図6に示す矩形31内の画素について同時に処理を行なう場合に、三角形30の外に位置する画素についての処理は行なわない。すなわち、図4に示す矩形31内の画素についての演算を行なっている間は、演算サブブロック 200_1 、 200_2 、 200_3 、 200_5 、 200_6 、 201_1 、 201_2 、 201_3 、 201_5 、 201_6 、 202_1 、 202_2 、 202_3 、 202_5 、 202_6 、 204_1 、 204_2 、 204_3 、 204_5 、 204_6 、 205_1 、 205_2 、 205_3 、 205_5 、 205_6 は停止した状態になり、これらの演算サブブロックは電力を消費しない。

【0044】、

以上説明したように、3次元コンピュータグラフィックシステム1によれば、テクスチャエンジン回路12におけるパイプライン処理において、同時処理する8画素のうち、処理対象となる三角形の外部に位置する画素についての演算は行なわないようにすることができる。

そのため、テクスチャエンジン回路12における消費電力を大幅に低減できる。その結果、3次元コンピュータグラフィックシステム1の電源として、簡単かつ安価なものをを用いることができる。

なお、テクスチャエンジン回路12は、図3および図4に示すように、各演算サブブロックに、クロックイネーブラおよび1ビットのフラグ用フリップフロップを組み込むことで、上述した機能を実現するが、クロックイネーブラおよび1ビットのフラグ用フリップフロップの回路規模は小さいため、テクスチャエンジン回路12の回路規模が大幅に増大することはない。

【0045】

本発明は上述した実施形態には限定されない。

例えば、上述した実施形態では、図3に示すように、テクスチャエンジン回路12およびメモリI/F回路13におけるパイプライン処理を行なう演算サブブ

ロックについて、 val データ $S220_1 \sim S220_8$ を利用する場合を例示したが、例えば、図1にレンダリング回路5内のDDAセットアップ回路10、トライアングルDDA回路11、テクスチャエンジン回路12およびメモリI/F回路13における処理のうち、パイプライン処理を行なわない所定の処理について、図5に示すように、 val データ $S320_1 \sim S320_8$ を用いて、演算処理の実行の有無を決定するようにしてもよい。

【0046】

また、上述した図1に示す3次元コンピュータグラフィックシステム1では、SRAM17を用いる構成を例示したが、SRAM17を設けない構成にしてもよい。

また、図1に示すテクスチャバッファ20およびテクスチャCLUTバッファ23を、DRAM16の外部に設けてもよい。

【0047】

また、上述した3次元コンピュータグラフィックシステム1では、3次元画像を表示する場合を例示したが、本発明は複数画素についてのデータを同時に処理して2次元画像を表示する場合にも適用できる。

また、上述した実施形態では、図2に示すように、画像処理の対象となる(z , R , G , B , α , s , t , q)データに、有効指示データとしての val データを付加したDDAデータ $S11$ を用いた場合を例示したが、(z , R , G , B , α , s , t , q)データと、 val データとを別個独立のデータとして扱うようにしてもよい。

【0048】

また、図3に示す3次元コンピュータグラフィックシステム1では、ポリゴンレンダリングデータを生成するジオメトリ処理を、メインプロセッサ4で行なう場合を例示したが、レンダリング回路5で行なう構成にしてもよい。

【0049】

さらに、上述した実施形態では、単位図形として三角形を例示したが、単位図形は特に限定されず、例えば、矩形であってもよい。

【0050】

【発明の効果】

以上説明したように、本発明のグラフィック演算装置およびその方法によれば、消費電力の大幅な低下を図ることができる。

そのため、本発明のグラフィック演算装置によれば、小規模かつ簡単な構成の電源を用いることができ、小規模化を図れる。

【図面の簡単な説明】

【図1】

図1は、本発明の実施形態の3次元コンピュータグラフィックシステムのシステム構成図である。

【図2】

図1に示すトライアングルDDA回路から出力されるDDAデータのフォーマットを説明するための図であう。

【図3】

図3は、図1に示すテクスチャエンジン回路およびメモリI/F回路の部分構成図である。

【図4】

図4は、図3に示す演算サブブロックの内部構成図である。

【図5】

図5は、図1に示す3次元コンピュータグラフィックシステムにおけるクロックイネーブラーを適用した、パイプライン処理を行っていない演算ブロックの構成図である。

【図6】

図6は、従来技術の問題点を説明するための図である。

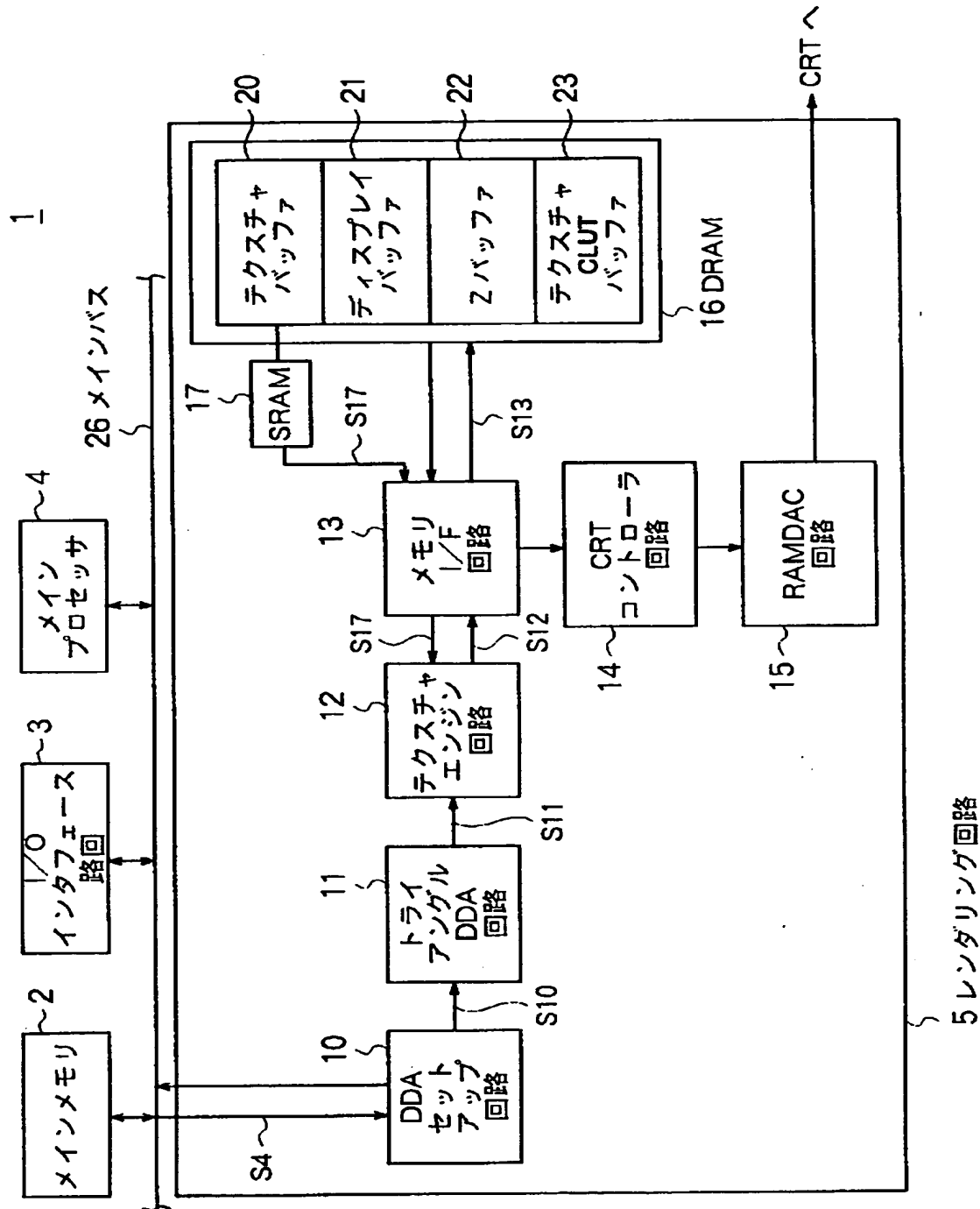
【符号の説明】

1…3次元コンピュータグラフィックシステム、2…メインメモリ、3…I/Oインタフェース回路、4…メインプロセッサ、5…レンダリング回路、10…DDAセットアップ回路、11…トライアングルDDA回路、12…テクスチャエンジン回路、13…メモリI/F回路、14…CRTコントローラ回路、15

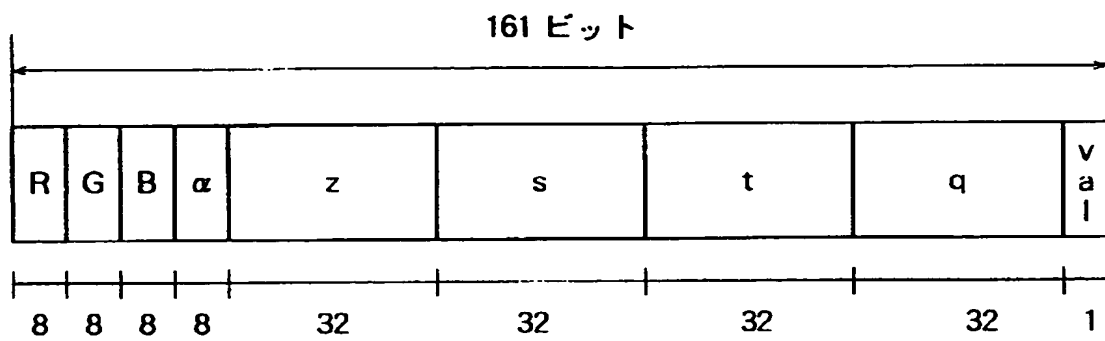
…RAMDAC回路、16…DRAM、17…SRAM、20…テクスチャバッファ、21…ディスプレイバッファ、22…Zバッファ、23…テクスチャCLUTバッファ、200～205…演算ブロック、 $200_1 \sim 200_8$ ， $201_1 \sim 201_8$ ， $202_1 \sim 202_8$ ， $203_1 \sim 203_8$ ， $204_1 \sim 204_8$ ， $205_1 \sim 205_8$ …演算サブブロック、 $210_1 \sim 210_8$ ， $211_1 \sim 211_8$ ， $212_1 \sim 212_8$ ， $213_1 \sim 213_8$ ， $214_1 \sim 214_8$ ， $215_1 \sim 215_8$ …クロックイネーブラ、222…データ用フリップフロップ、223…プロセッサエレメント、224…フラグ用フリップフロップ

【書類名】 図面

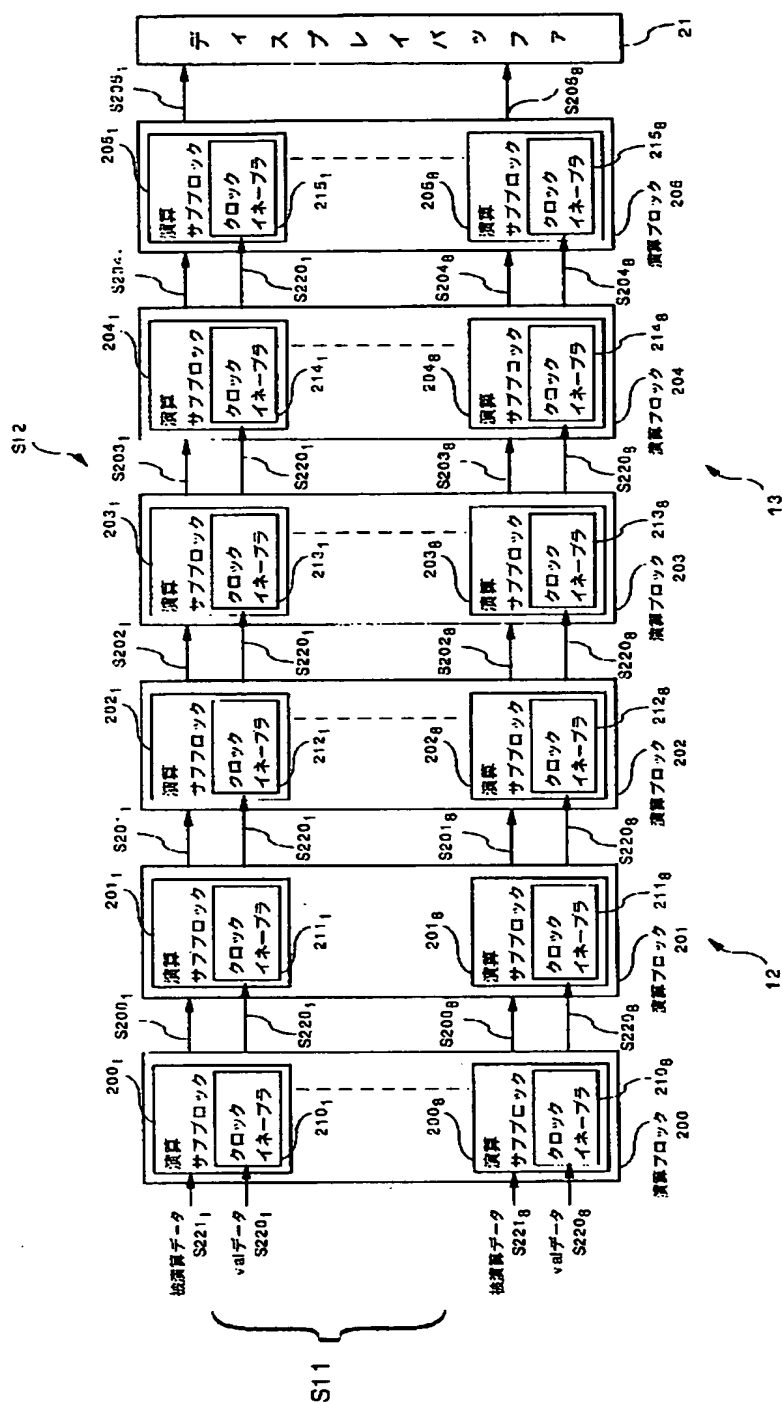
【図 1】



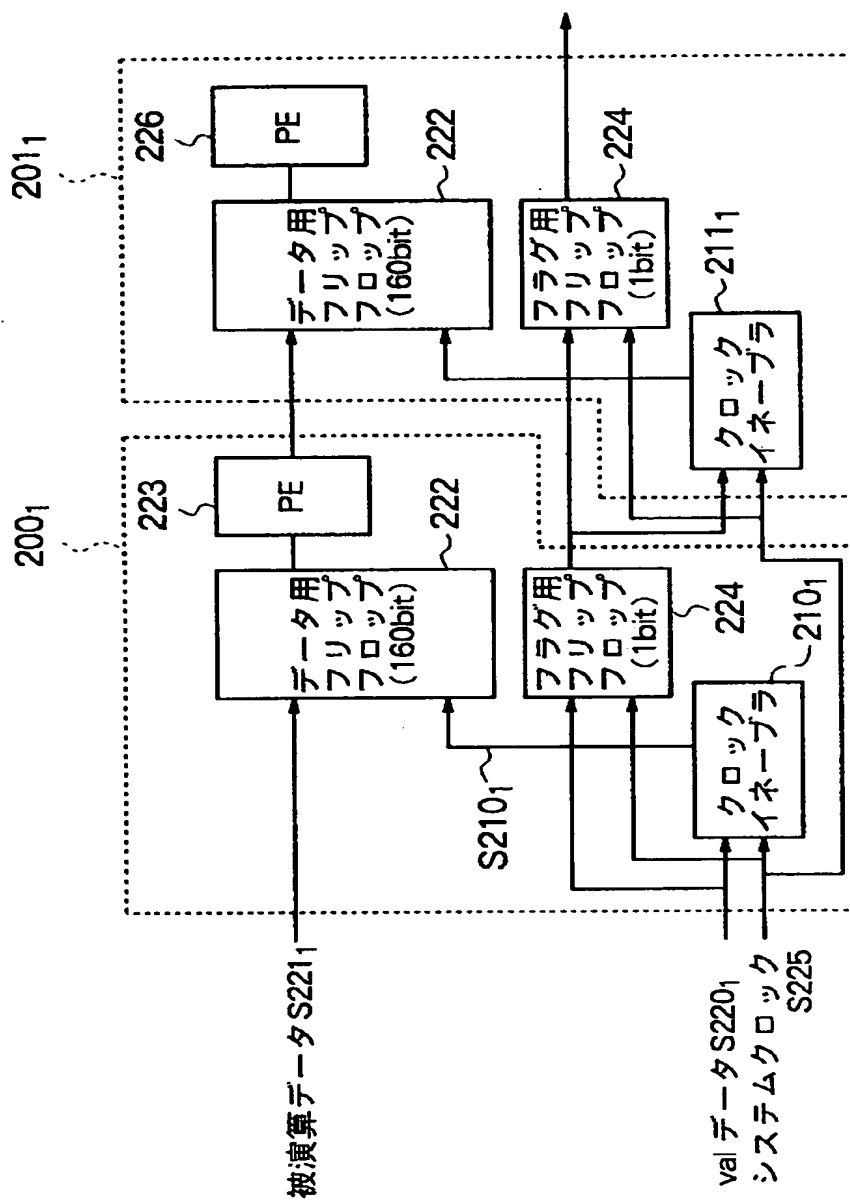
【図 2】



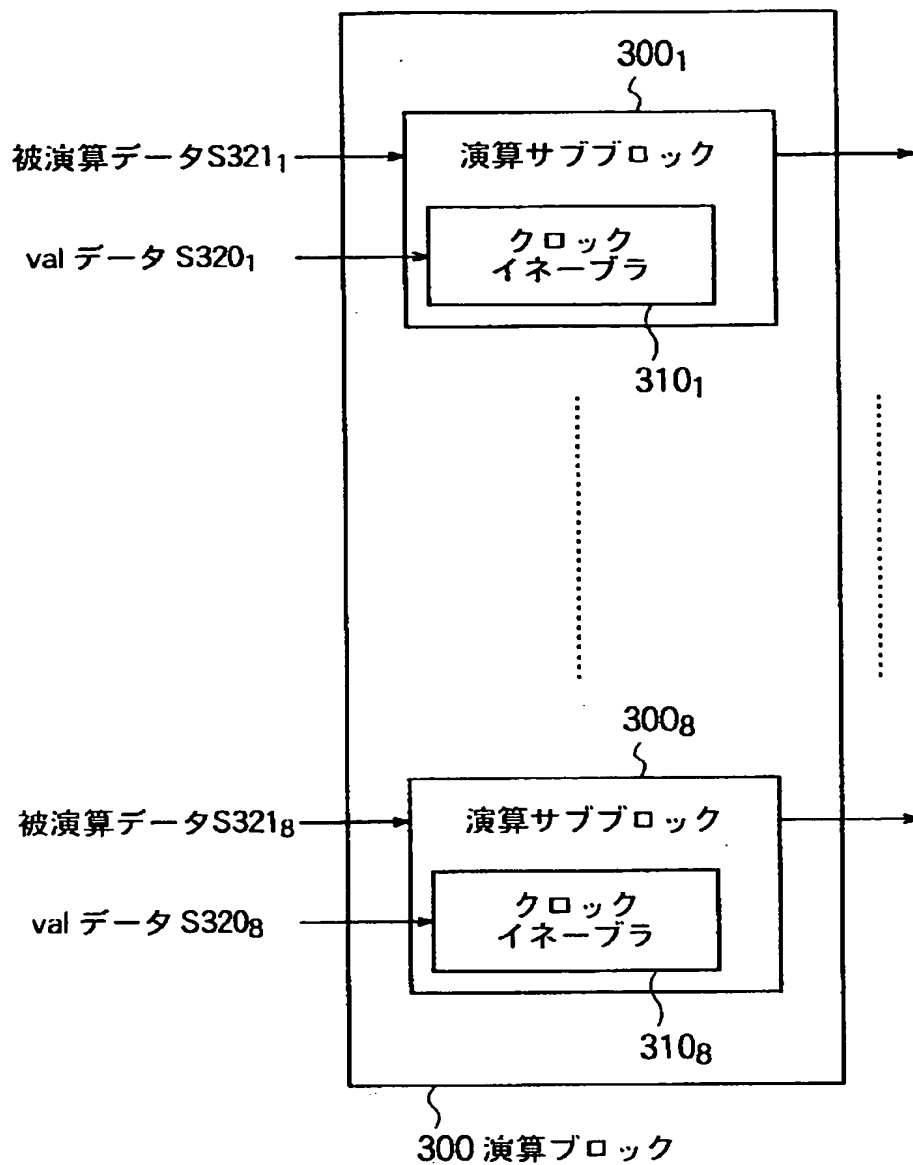
【図 3】



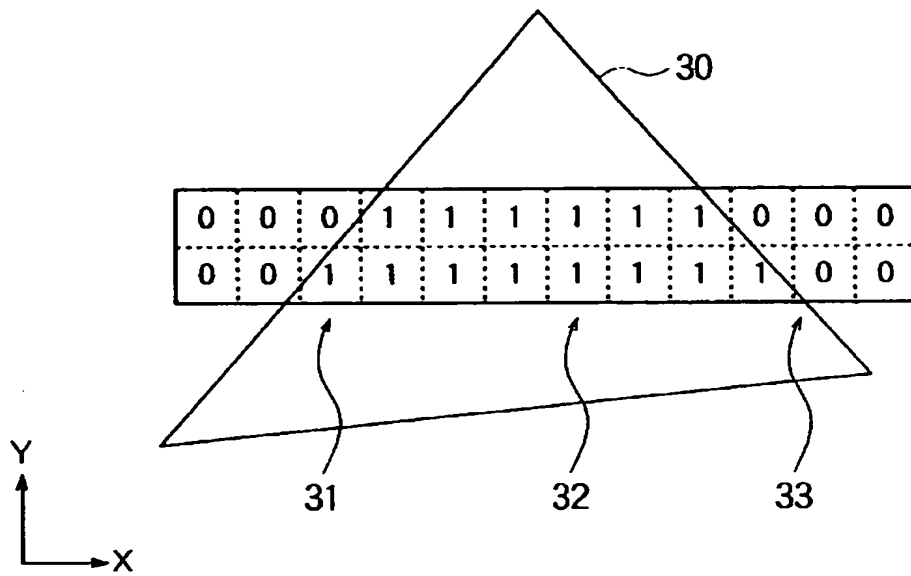
【図 4】



【図 5】



【図 6】



【書類名】 要約書

【要約】

【課題】 消費電力の大幅な低下を図れるグラフィック演算装置を提供する

【解決手段】 ディスプレイに表示する所定の形状を単位図形の組み合わせで表現するために、複数の画素についての演算を同時に行い、処理対象となっている前記単位図形の内側に位置する画素についての演算結果を有効なものとして用いて処理を行なうグラフィック演算装置であって、演算サブブロック $200_1 \sim 205_8$ において、それぞれ対応する val データ $S220_1 \sim S220_8$ の有効性がクロックイネーブラ $210_1 \sim 215_8$ で判断され、対応する val データが有効を示す演算サブブロックのみが演算を実行し、そうでない演算サブブロックは演算を行なわない。演算ブロック $200 \sim 205$ は、パイプライン処理を実現する。

【選択図】 図3

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】
【識別番号】 000002185
【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号
【氏名又は名称】 ソニー株式会社
【代理人】 申請人
【識別番号】 100094053
【住所又は居所】 東京都台東区柳橋 2 丁目 4 番 2 号 創進国際特許事務所
【氏名又は名称】 佐藤 隆久

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日	1990年 8月30日
[変更理由]	新規登録
住 所	東京都品川区北品川6丁目7番35号
氏 名	ソニー株式会社